AD

REPORT NO. RG-TR-68-12

LARGE-SCALE INTEGRATED CIRCUIT COMPUTER FOR COORDINATE TRANSFORMATION

by

Raul Pena, Jr.

August 1968

This document is subject to special export controls and each transmittal to foreign governments or foreign nationals may be made only with prior approval of this Command, ATTN. AMSMI-RG.

U.S. ARMY MISSILE COMMAND
Redstone Arsenal, Alabama

FORM AMSMI-1021, 1 DEC 65 PREVIOUS EDITION IS OBSOLETE

37

12 August 1968

Report No. RG-TR-68-12

LARGE-SCALE INTEGRATED CIRCUIT COMPUTER FOR COORDINATE TRANSFORMATION

by

Raul Pena, Jr.

DA Project No. 1X27919D678
AMC Management Structure Code No. 5282.12.12700

Guidance System Branch
Army Inertial Guidance and Control Laboratory and Center
Research and Development Directorate
U. S. Army Missile Command
Redstone Arsenal, Alabama 35809

ABSTRACT

Methods and algorithms for navigation systems with body-mounted inertial instruments have been studied by various research groups throughout the country. The high iteration rates for coordinate transformation in a PERSHING II missile and the complexity of the navigation solutions make a general purpose computer or special purpose computer unfeasible for this application. A scheme to use a small Digital Differential Analyzer as the front end of a general purpose navigation computer to perform the coordinate transformation is feasible.

This report describes a unique mechanization of a Digital Differential Analyzer to transform the outputs from inertial instruments from a body-fixed coordinate system to a stabilized coordinate system. The design is specially adaptable to various levels of logic density using large-scale integrated circuits because it has few different logic circuits used repeatedly and performs serial computation that requires minimum input and output leads. Unique features of the Digital Differential Analyzer are that it solves a 3×3 direction cosine matrix by merely circulating data around a pair of registers and adders for each matrix element and uses a two-bit adder to add, subtract, and multiply by $(2)^n$.

CONTENTS

		Page
1.	Introduction	1
2.	Description	5
3.	Sequencing and Operations	5
4.	Interface With Navigation Computer	19
5.	Hardware Design	19
6.	Input/Output	21
7.	Hardware Construction	21
8.	Nuclear Effect Considerations	22
9.	Conclusion	22
Appendix.	FLOW OF DATA IN COMPLETE COMPUTER FUNCTION FOR TWO UPDATINGS	25

ILLUSTRATIONS

Table

I	Equivalents of Binary and Decimal Systems	7

Figure

1	Flow of Data for Two Updatings of Direction Cosines for Row One	2
2	Block Diagram of the Digital Differential Analyzer Showing Preset Values of Directional Cosines in Preset Position	6
3	Comparison of the Most and Least Significant Parts of the Outputing Matrix Element	9
4	Timing Period 0	11
5	Timing Period 1	13
6	Timing Period 2	14
7	Timing Period 3	16
8	Timing Period 4	17
9	Timing Period 5	18

1. Introduction

Various research groups throughout the country have made feasibility studies and developed algorithms for the computations required for a system with body-mounted inertial sensors (strapdown). The Systems Branch of the Army Inertial Guidance and Control Laboratory and Center has also made such studies and established requirements, computational speeds and accuracies, and selected algorithms for a computer to perform those computations.

The high iteration rates for computational accuracy of those algorithms in a PERSHING II application plus the complexity of computation of the navigation equations make the use of a general purpose computer or special purpose computer unfeasible. A scheme to use a small, fast, Digital Differential Analyzer which employs fast, low-powered, compact, large-scale integrated circuits to compute the direction cosine algorithms was selected. This Digital Differential Analyzer will transform the outputs of inertial sensors from a body-fixed coordinate system to a stabilized coordinate system and feed them to a simple navigation computer for solution of the balance of the guidance and control problem.

This report describes the mechanization of a Digital Differential Analyzer designed to be built from large-scale integrated circuits to perform the coordinate transformation. More precisely, this Digital Differential Analyzer was designed to transform the outputs of body-mounted gyroscopes and accelerometers for missiles up to a PERSHING II class in a strapdown application into equivalent outputs from a stabilized gimbal system. Beyond that navigation and guidance problem remains unchanged.

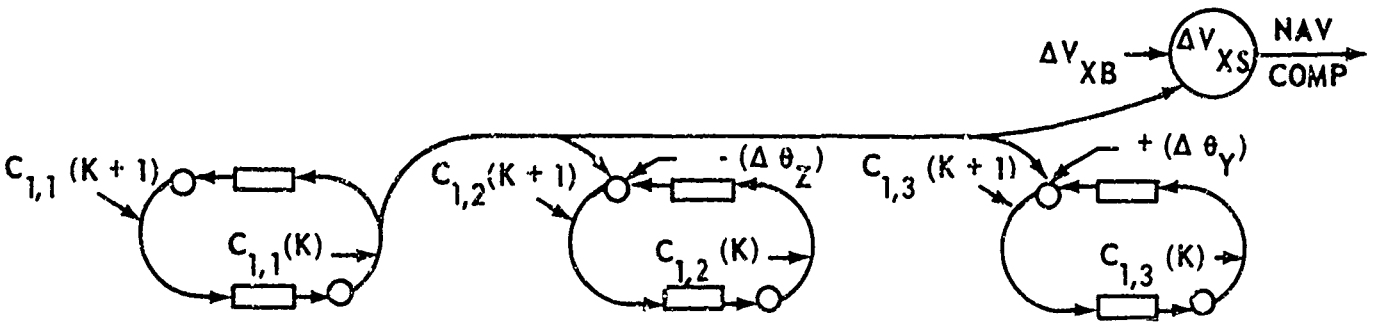
The algorithms and organization for this transformation are described in another report.[1] The scheme to mechanize those algorithms is also described in another report.[2] This latter report describes the various schemes considered and tradeoffs effected, the type of outputs from the inertial instruments, and the speeds required to maintain computational accuracy required for such a missile system.

---

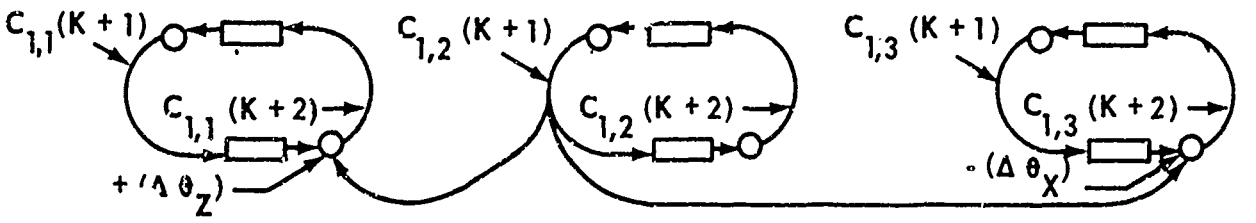
[1]U.S. Army Missile Command, Redstone Arsenal, Alabama, Digital Differential Analyzer Organization for an Analytic Platform Transform Computer by Gerald Scheiman, January 1967, Report No. RG-TR-67-2.

[2]U.S. Army Missile Command, Redstone Arsenal, Alabama, Pershing Semi-Annual Report, Guidance Systems Branch by M. C. Jones, et al., 8 February 1968, Report No. ITN-RGG-18.

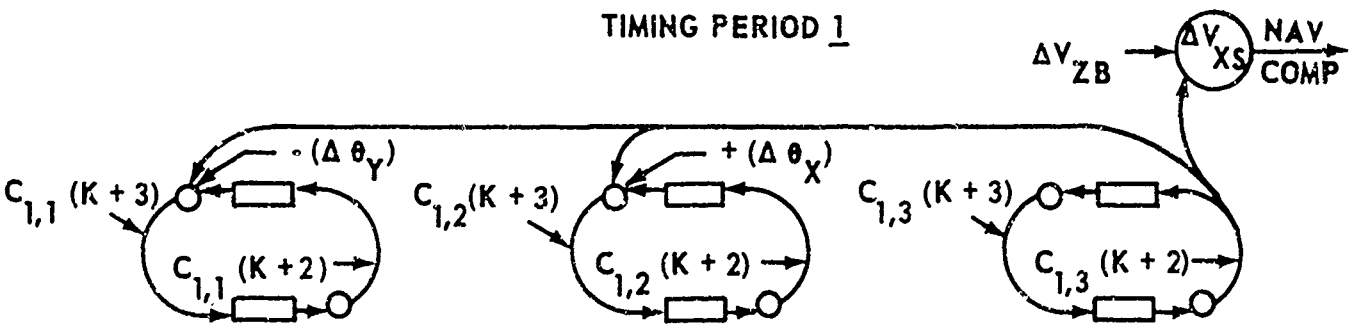
The flow and sequence of data representing the two updatings of the algorithms given below are shown in Figure 1. The complete computer has three rows operating simultanecusly. A more detailed description of each updating appears later in this report. The appendix gives the flow of data.



TIMING PERIOD 0



TIMING PERIOD 1



TIMING PERIOD 2

UPDATING 1

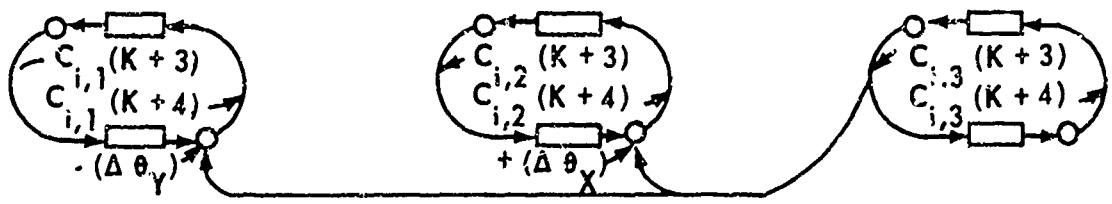
ΔV XS ΔV ACCUMULATOR FOR ROW ONE

O ADDER

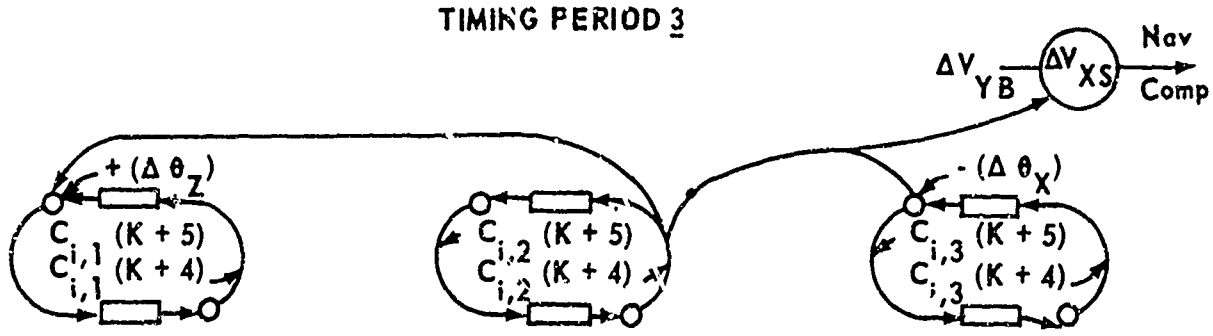
REGISTER

NOTE: ROWS TWO AND THREE ARE IDENTICAL

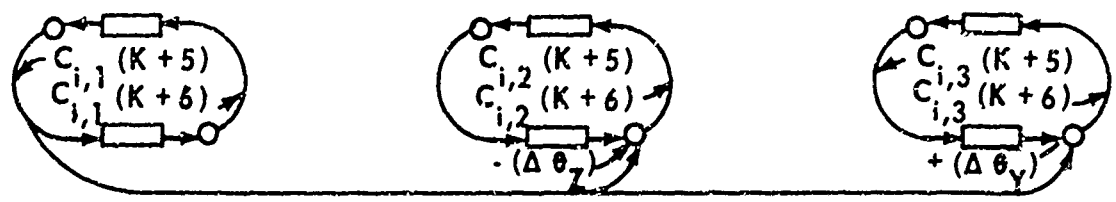
FIGURE 1. FLOW OF DATA FOR TWO UPDATINGS OF DIRECTION COSINES FOR ROW ONE



TIMING PERIOD 3



TIMING PERIOD 4



TIMING PERIOD 5

UPDATING 2

NOTE: SEE APPENDIX FOR FLOW OF DATA IN COMPLETE COMPUTER FOR TWO UPDATINGS

FIGURE 1. FLOW OF DATA FOR TWO UPDATINGS OF DIRECTION COSINES FOR ROW ONE (Concluded)

The algorithms are:

$$C_{i,1}(K+1) = C_{i,1}(K)$$
$$C_{i,2}(K+1) = C_{i,2}(K) - C_{i,1}(K)\,\Delta\theta_Z$$
$$C_{i,3}(K+1) = C_{i,3}(K) + C_{i,1}(K)\,\Delta\theta_Y$$

$$C_{i,1}(K+2) = C_{i,1}(K+1) + C_{i,2}(K+1)\,\Delta\theta_Z$$
$$C_{i,2}(K+2) = C_{i,2}(K+1)$$
$$C_{i,3}(K+2) = C_{i,3}(K+1) - C_{i,2}(K+1)\,\Delta\theta_X$$

$$C_{i,1}(K+3) = C_{i,1}(K+2) - C_{i,3}(K+2)\,\Delta\theta_Y$$
$$C_{i,2}(K+3) = C_{i,2}(K+2) + C_{i,3}(K+2)\,\Delta\theta_X$$
$$C_{i,3}(K+3) = C_{i,3}(K+2)$$

Timing period: 20 clock pulses each — 0

Updating 1 — 1

2

$$C_{i,1}(K+4) = C_{i,1}(K+3) - C_{i,3}(K+3)\,\Delta\theta_Y$$
$$C_{i,2}(K+4) = C_{i,2}(K+3) + C_{i,3}(K+3)\,\Delta\theta_X$$
$$C_{i,3}(K+4) = C_{i,3}(K+3)$$

3

$$C_{i,1}(K+5) = C_{i,1}(K+4) + C_{i,2}(K+4)\,\Delta\theta_Z$$
$$C_{i,2}(K+5) = C_{i,2}(K+4)$$
$$C_{i,3}(K+5) = C_{i,3}(K+4) - C_{i,2}(K+4)\,\Delta\theta_X$$

Updating 2 — 4

$$C_{i,1}(K+6) = C_{i,1}(K+5)$$
$$C_{i,2}(K+6) = C_{i,2}(K+5) - C_{i,1}(K+5)\,\Delta\theta_Z$$
$$C_{i,3}(K+6) = C_{i,3}(K+5) + C_{i,1}(K+5)\,\Delta\theta_Y$$

5

where

i = 1, 2, and 3 (for the three strings)

$C_{i,1}$ (K) = value of cosine matrix element

(K) = the K^{th} cycling of the total algorithm, updatings 1 and 2

$\Delta\theta_X$, $\Delta\theta_Y$, $\Delta\theta_Z$ = incremental outputs from the body-mounted gyros.

Algorithms are for solving 3 × 3 direction cosine matrix. Only one of the three strings is shown in two successive updatings. The other two strings are identical and the updating sequence repeats.

2. Description

The block diagram (Figure 2) represents the complete Digital Differential Analyzer and interface with the test computer and the navigation computer. There are nine matrix elements. The hardware for each of the matrix elements is identical, each being made up of one pair of adders and one pair of 16-bit registers. They are arranged in three identical columns and three rows. At any given time, the operation of every row is identical and the columns differ only by the number of clock pulses or by whether they are outputing or inputing. The sequencer and I/O provide the control functions. The following discussions will be limited to only one row since the other two are physically identical. The clocking for any column is the same for any timing period and is applied to each pair of adders simultaneously.

The computer uses a 2's complement number system in which negative numbers (sign bit = 1) are the 2's complement of positive binary numbers (sign bit = 0). This makes it possible to use a simple adder to perform addition and subtraction and automatically get the correct sign in the result. In subtraction the computer automatically forms the 2's complement of the subtrahend and adds. A short explanation of this system is given in Table I.

3. Sequencing and Operations

a. Organization

The studies and trade-offs in the previously mentioned report[3] established the word length for direction cosine to be 16 bits plus sign and the

---

[3]Jones, loc. cit.

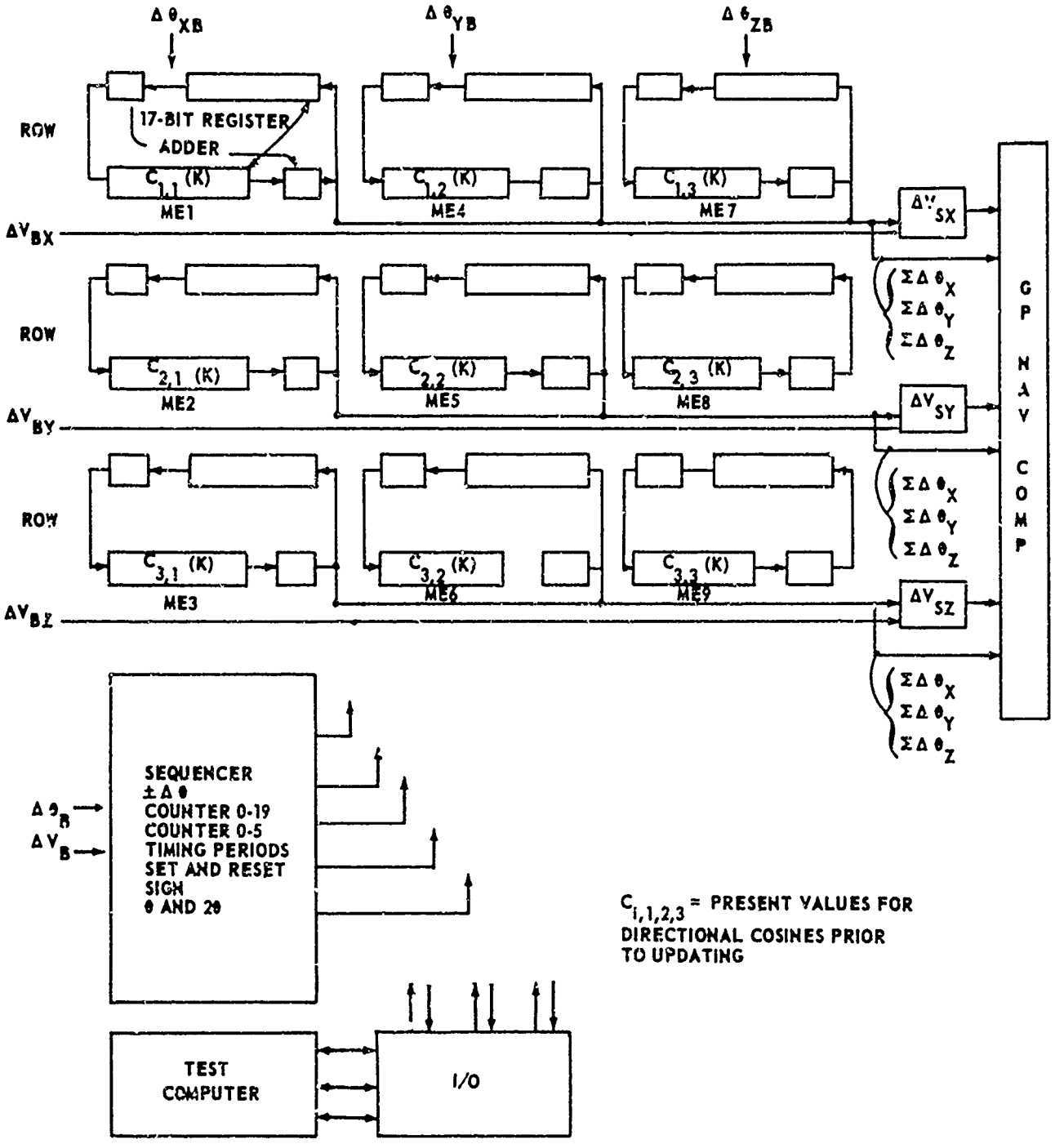


FIGURE 2. BLOCK DIAGRAM OF THE DIGITAL DIFFERENTIAL ANALYZER SHOWING PRESET VALUES OF DIRECTIONAL COSINES IN PRFSET POSITION

TABLE I. EQUIVALENTS OF BINARY AND DECIMAL SYSTEMS

BINARY	DECIMAL	BINARY	DECIMAL
01111	15	10001	-15
01110	14	10010	-14
01101	13	10011	-13
01100	12	10100	-12
01011	11	10101	-11
01010	10	10110	-10
01001	9	10111	- 9
01000	8	11000	- 8
00111	7	11001	- 7
00110	6	11010	- 6
00101	5	11011	- 5
00100	4	11100	- 4
00011	3	11101	- 3
00010	2	11110	- 2
00001	1	11111	- 1
00000	0		

NOTE: To form 2's complement (negative form) of a binary number, start from the least significant end; copy all 0 bits and the first bit; then complement all the rest.

Example: Binary 10 01010
2's complement 10110

An example of 2's complement arithmetic may be shown:

a. (+5) + (+7) = +12

+	+5	00101
	+7	00111
	+12	01100

b. (+5) + (−7) = −2

$$\begin{array}{rl} & +5 = 00101 \\ + & -7 = 11001 \\ \hline & -2 = 11110 \end{array}$$

c. (+5) − (−7) = +5 +7 = +12

$$(-)\ \begin{array}{ll} +5 & 00101 \\ -7 & 11001 \end{array} = \begin{array}{rl} & +5 = 00101 \\ + & +7 = 00111 \\ \hline & +12 = 01100 \end{array}$$

d. (−5) − (+7) = −5 −7 = −12

$$(-)\ \begin{array}{ll} -5 & 11011 \\ +7 & 00111 \end{array} = \begin{array}{rl} & -5 = 11011 \\ & -7 = 11001 \\ \hline & -12 = 10100 \end{array}$$

basic clock to be 1.2 mHz. The direction cosines of the $\Delta\theta_X$ column partially increment the $\Delta\theta_Y$ and $\Delta\theta_Z$ columns in one word time; the $\Delta\theta_Y$ column partially increments the $\Delta\theta_X$ and $\Delta\theta_Z$ columns and the $\Delta\theta_Z$ column partially increments the $\Delta\theta_X$ and $\Delta\theta_Y$ columns (Algorithms list, paragraph 1). Complete updating is obtained in three word times, after which the sequence is reversed to increase numerical accuracy of the computation when limit cycle angular inputs are received.[4] The direction cosines from each column are combined with appropriate corresponding ΔV_B (body axis) to convert them to ΔV_S (space axis).

b. Scaling and Updating of Direction Cosines

The gyros are scaled so that an angular increment of $\Delta\theta$ will output one pulse and increment the direction cosine by a maximum of one bit in the bit 16 position. Since the direction cosines update each other, the matrix elements were so sized that each direction cosine word is actually 31 bits plus sign in length, and these are held in pairs of 16-bit registers that are clocked simultaneously during operation. During the first word period the most significant part of the $\Delta\theta_X$ column is added to the least significant part of the $\Delta\theta_Y$ and $\Delta\theta_Z$ columns. The overflow resulting from such addition is the carry left in the respective adder and is automatically added to the most

[4]Scheiman, loc. cit.

significant part of the $\Delta\theta_Y$ and $\Delta\theta_Z$ columns to increment them partially. This is a very convenient arrangement, since the adder is the only place where a bit can be temporarily stored. The next timing period of the register will shift and pick up the stored carry. Similarly, during the second word time, the $\Delta\theta_Y$ matrix element will partially update the $\Delta\theta_X$ and $\Delta\theta_Z$ matrix elements, and during the third word time, the $\Delta\theta_Z$ matrix element matrix will partially update the $\Delta\theta_X$ and $\Delta\theta_Y$ matrices element, thus completing one updating of direction cosines in one row. The same occurs simultaneously in the other two rows. The maximum accelerometer output occurs less than half as often as the gyros and are sampled to be combined with direction cosines every other updating period.

c. Sequencing

The updating arrangement is straightforward, but mechanization is complex because of the unequal shifting that is necessary at every column during every word time and the change in scale factor of the gyros for outputs during maximum body rates. Figure 3 illustrates how the most significant part of the outputing matrix element combines with the least significant part of the matrix element which is partially incremented. The sign bit is added to all bits until it coincides with the sign bit of the matrix element being updated. There is a slight overlap, and the matrix element that is to output next must stop when bit 16 is the next bit through the adder. This uneven clocking requires overlapping to complete any partial updating. The overlapping occurs during the next word time when the next partial updating takes place. Therefore, it is necessary to keep track of $\Delta\theta$, sign bit, bit 31, carry, and carry reset. The direction cosine at the top is updated by the direction cosine at the bottom.

The pulse rebalanced gyros being considered for this application will double the scale factor when body rate exceeds a certain level in that axis.

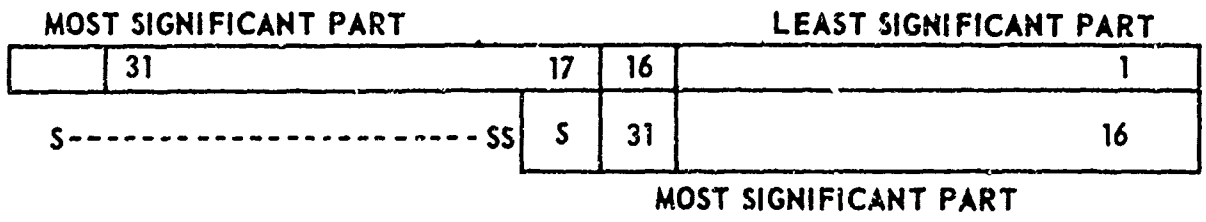


FIGURE 3. COMPARISON OF THE MOST AND LEAST SIGNIFICANT PARTS OF THE OUTPUTING MATRIX ELEMENT

Pulses from the gyro will then represent $2\Delta\theta$, and the direction cosines must be updated twice as much or twice as fast as those for the $\Delta\theta$ case. In most cases, this change in scale factor will not occur in all three axes at the same time, so the choice of doubling iteration rates at random for one, two, or all three axes is not feasible. This is analogous to shifting gears on a tricycle but affecting only one, two, or sometimes all three wheels; of course, the machine could be made to operate always twice as fast as in the worst case and provide logic to sample gyro pulses every other time for the $\Delta\theta$ case. This approach, however, would unnecessarily increase logic speed. Logic for multiplying, even by two, is much more complex than the plain two-bit adders being used. The mechanization implemented for the $2\Delta\theta$ was to double the size of the updating direction cosine by running it through a flip-flop, causing a one-clock period delay as it entered the input adder of the matrix element being updated. In binary numbers this is the same as shifting the whole word one bit to the left, which is alternately the same as multiplying by two. The solution was simple, but logic had to be added for the computer to recognize the $2\Delta\theta$ case and delay the word without changing the number of clock pulses for the $\Delta\theta$ case and retain the information during the overlap portion of updating. The word time for each register will vary from 15 to 17 clock pulses. Two successive updatings are accomplished in six timing periods (0 to 5) of twenty clock pulses (t0 to t19) each. The sequencing stops each register at the appropriate time. The following sequence describes what takes place in each matrix element of each row during every timing period. It is based on the assumption that $\Delta\theta$ pulse is present in the matrix unit being updated as will be the case with binary pulse rebalance gyros. The effect of zero angular input to the gyro will be that the gyro outputs and algorithms will alternate plus and minus for a net direction cosine increment of zero.

(1) Timing Period 0 (Figure 4)

(a) Overlap. During previous timing period 5 adder A2 outputs bits 16 to 31 into matrix element 4 (ME4) and ME7 through A3 and A5, respectively. The sign bit was not added, so it is the first one out of register (R2) during this timing period. Therefore, the first clock period will allow the sign bit to go into A3 and A5 to set the sign to be added to bits 17 plus sign and complete the partial updating started during 5.

(b) Partial Updating This Period. Matrix element 1 will receive 17 clock pulses and output bits 16 to 31 plus sign through A1 to partially increment ME4 and ME7. Matrix element 4 will be the next to output through A4, so it will receive only 15 clock pulses and thus add only bits 16 to 30 from ME1 to bits 1 to 15. Matrix element 7 receives 16 clock pulses because it will

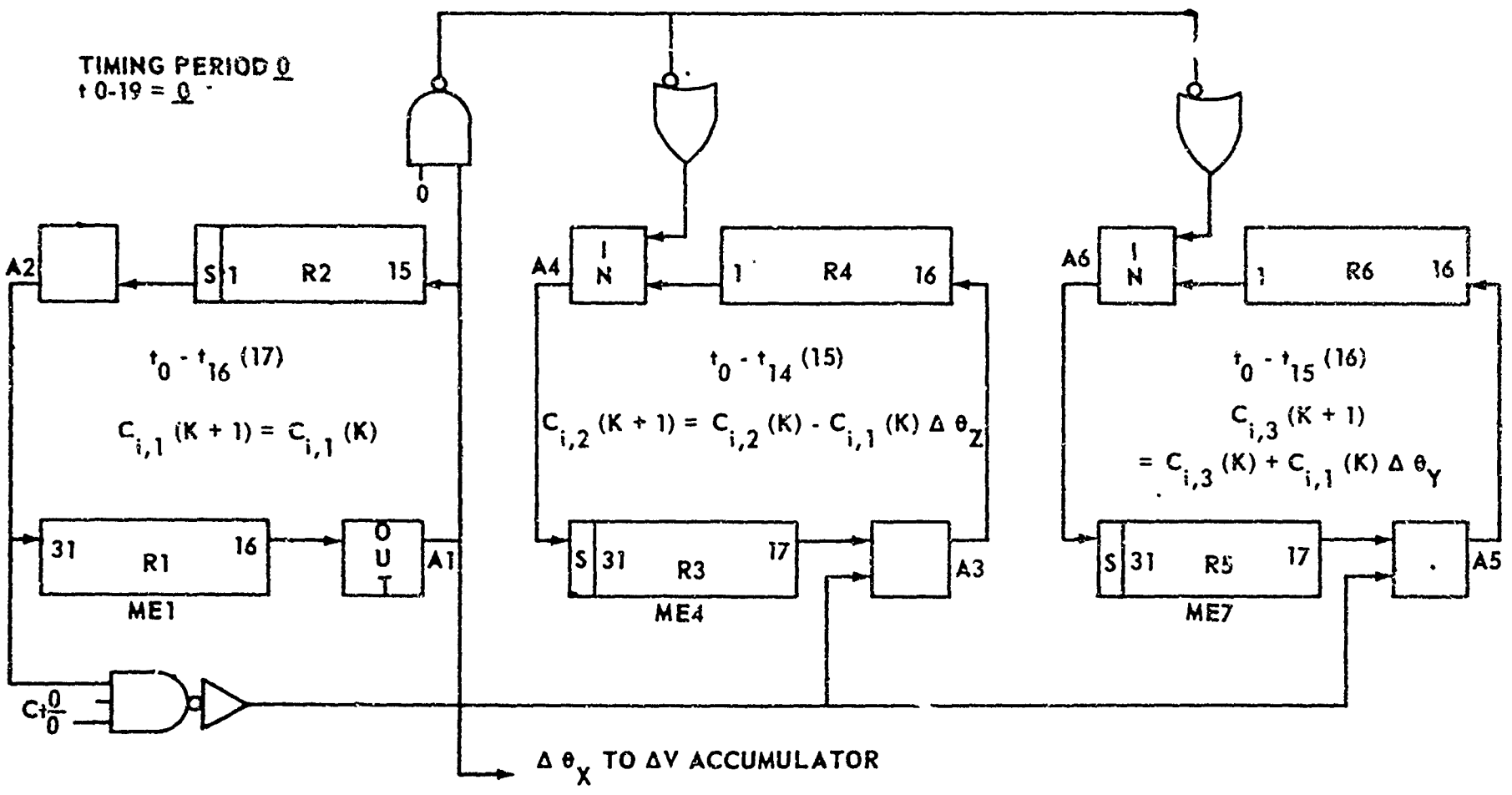


FIGURE 4. TIMING PERIOD 0

be updated through A5 during the next timing period and must have bit 1 as the next bit into A5. Bits 16 to 31 from ME1 are added to bits 1 to 16.

During this timing period, bits 16 to 31 plus sign are simultaneously outputed to the ΔV accumulator where they are combined with ΔV_{XB} to form the ΔV_X component of ΔV_{XS} ready to be sampled by the navigation computer.

(2) Timing Period 1 (Figure 5)

(a) Overlap. Bits 31 and sign from ME1 into ME4 and sign bit from ME1 into ME7 must be added to complete the partial updating of $\Delta\theta_Y$ and $\Delta\theta_Z$ columns from timing period 0. Therefore, the first two clock pulses of this period will add bit 31 and set sign in A4 of ME4 and the first clock pulse of this period will set sign in A6 of ME7.

(b) Partial Updating This Period. Matrix element 1 receives 16 clock pulses to place bit 1 as the next one into A2 since it will also receive data during the next timing period 2. Bits 16 to 31 from ME4 are added to bits 1 to 16 in A1. Matrix element 4 receives 17 clock pulses to output through A4 to partially increment ME1 through A1 and ME7 through A5. Matrix element 7 receives 15 clock pulses to stop bit 16 just before A5 because it will increment ME4 and ME1 during the next timing cycle. Bits 16 to 30 from A4 are added to bits 1 to 15 through A5.

(3) Timing Period 2 (Figure 6)

(a) Overlap. This overlap is the same as that of timing period 1 except that it includes bits 31 and sign from ME4 into ME7 and sign from ME4 into ME1.

(b) Partial Updating This Period. Matrix element 1 receives 16 clock pulses to place bit one as the next bit into A1 since it will also receive data during the next timing period 3. Bits 16 to 31 from ME7 are added to bits 1 to 16 in A4 of ME4, and in A2 of ME1.

During this timing period, bits 16 to 31 plus sign are simultaneously outputed to the ΔV accumulator where they are combined with ΔV_{ZB} to form the ΔV_Z component of ΔV_{XS}.

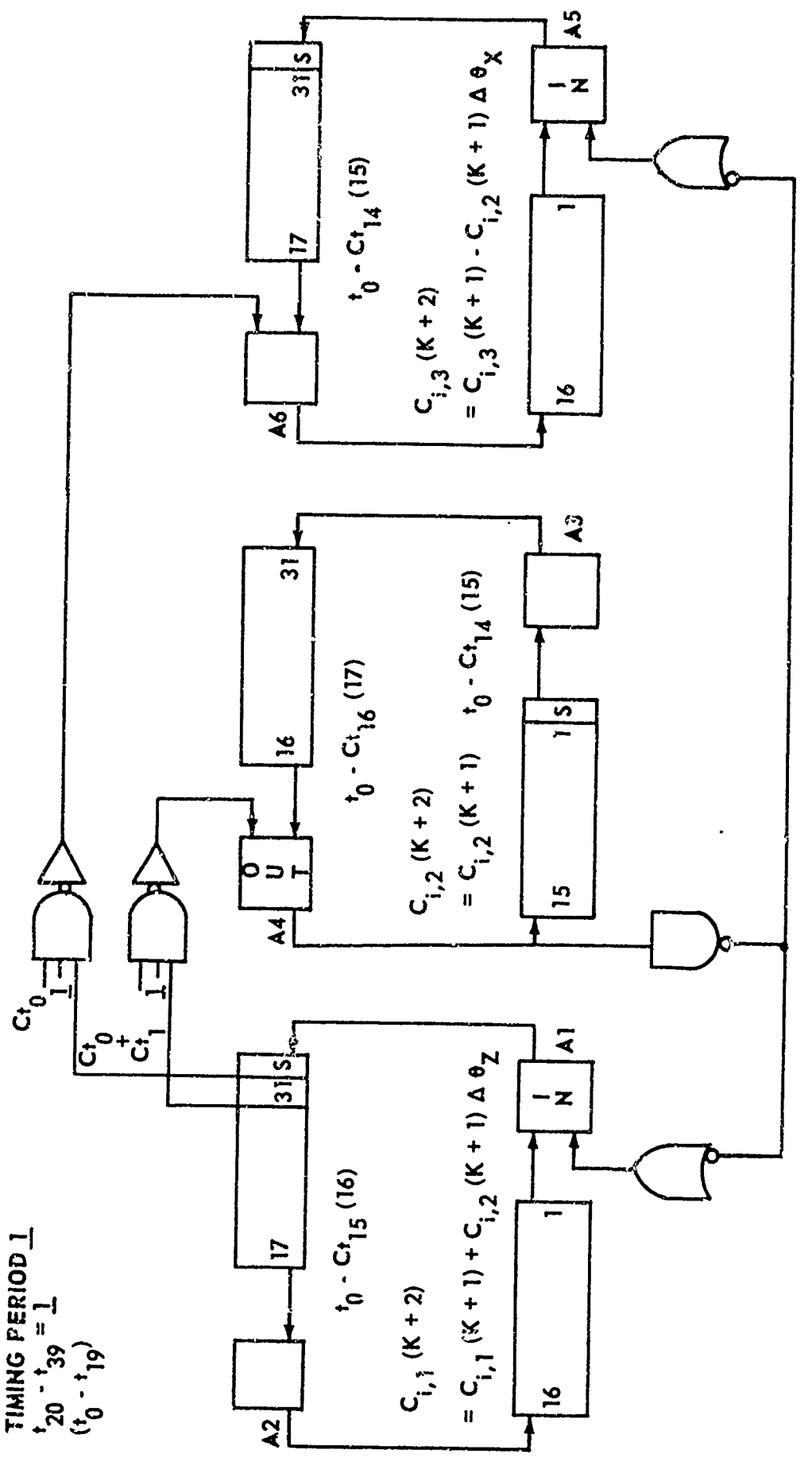


FIGURE 5. TIMING PERIOD 1

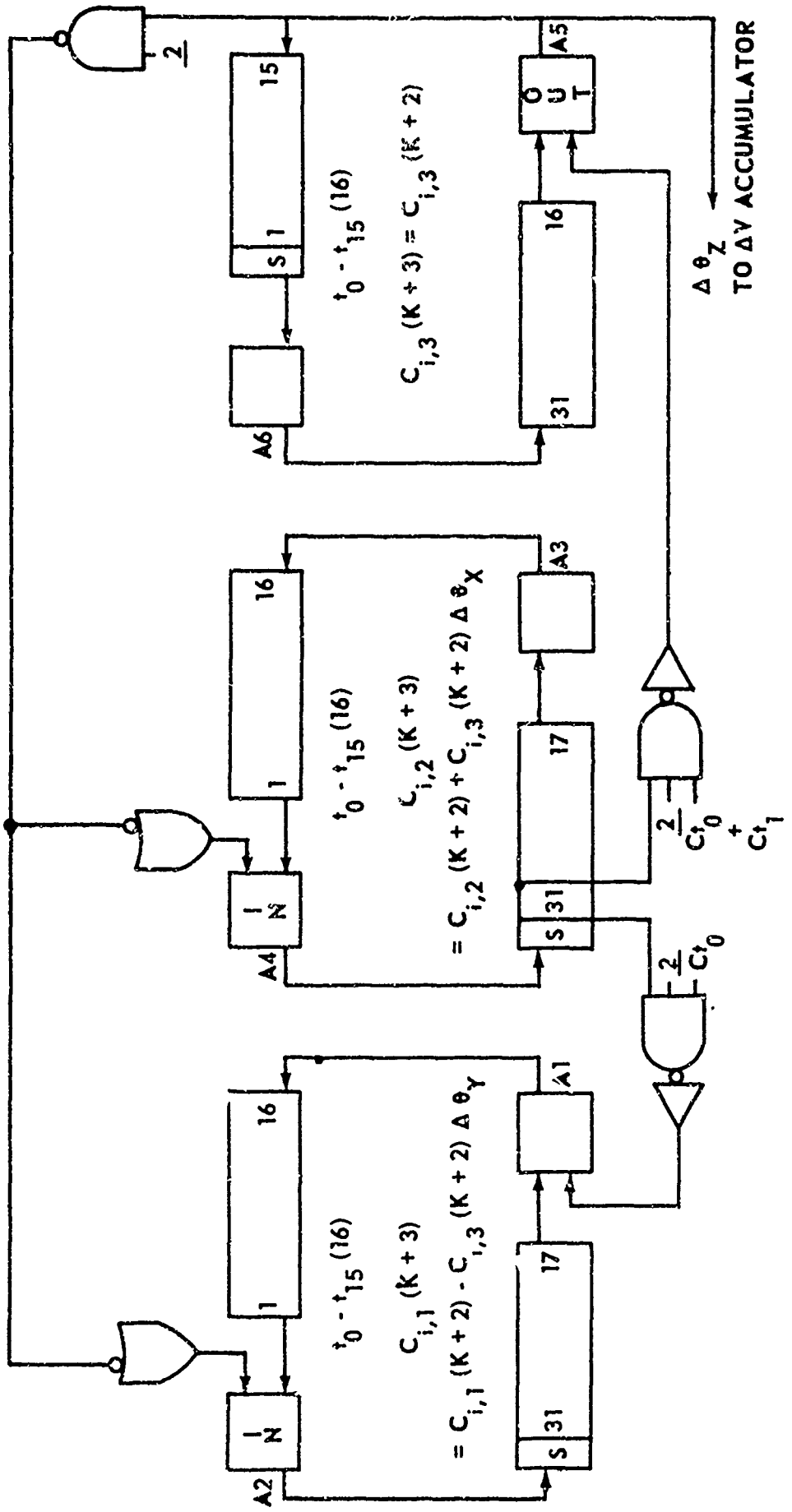


FIGURE 6. TIMING PERIOD 2

(4) Timing Period 3 (Figure 7)

(a) Overlap. The first clock pulse adds the sign bit from ME7 into ME4 and ME1 to complete the partial updating of $\Delta\theta_X$ and $\Delta\theta_Y$ from timing period 2.

(b) Partial Updating This Period. Matrix element 1 receives 16 clock pulses to place bit 1 as the next bit into A2 since it will also receive data during the next timing period 3. Bits 16 to 31 from ME7 are added to bits 1 to 16 of ME1 through A1. Matrix element 4 receives 15 clock pulses to place bit 16 as the next bit into A3. Bits 16 to 30 from ME7 are added to bits 1 to 15 of ME4 through A3. Matrix element 7 receives 17 clock pulses to output bits 16 to 31 plus sign and place bit 1 as the next bit into A6 since it will receive data during the next timing period.

(5) Timing Period 4 (Figure 8)

(a) Overlap. The overlap of timing period 4 is the same as that of timing period 1 except that it is bit 31 and sign from ME7 into ME4 and sign bit from ME7 into ME1.

(b) Partial Updating This Timing Period. Matrix element 1 receives 15 clock pulses to place bit 16 as the next bit into A2 since it will output data during the next timing period. Bits 16 to 30 from ME4 are added to bits 1 to 15 of ME1 through A2. Matrix element 4 receives 17 clock pulses to output bits 16 to 31 plus sign and put bit 1 as the next bit into A3 since it will receive data during the next timing period. Matrix element 7 receives 16 clock pulses to make bit 1 the next bit into A5 since it will receive data during the next timing period. Bits 16 to 31 from ME4 are added to bits 1 to 16 of ME7 through A6.

During this timing period, bits 16 to 31 plus sign are simultaneously outputed from ME4 to the ΔV accumulator where they are combined with ΔV_{YB} to form the ΔV_Y component of ΔV_{XS}.

(6) Timing Period 5 (Figure 9)

(a) Overlap. The first clock pulse adds the sign bit from ME4 into ME7 and bit 31 and sign from ME4 into ME1 to complete the partial updating of $\Delta\theta_X$ and $\Delta\theta_Z$ from timing period 4.

(b) Updating This Timing Period. Matrix element 1 receives 16 clock pulses to output bits 16 to 31 and place bit 16 as the next bit

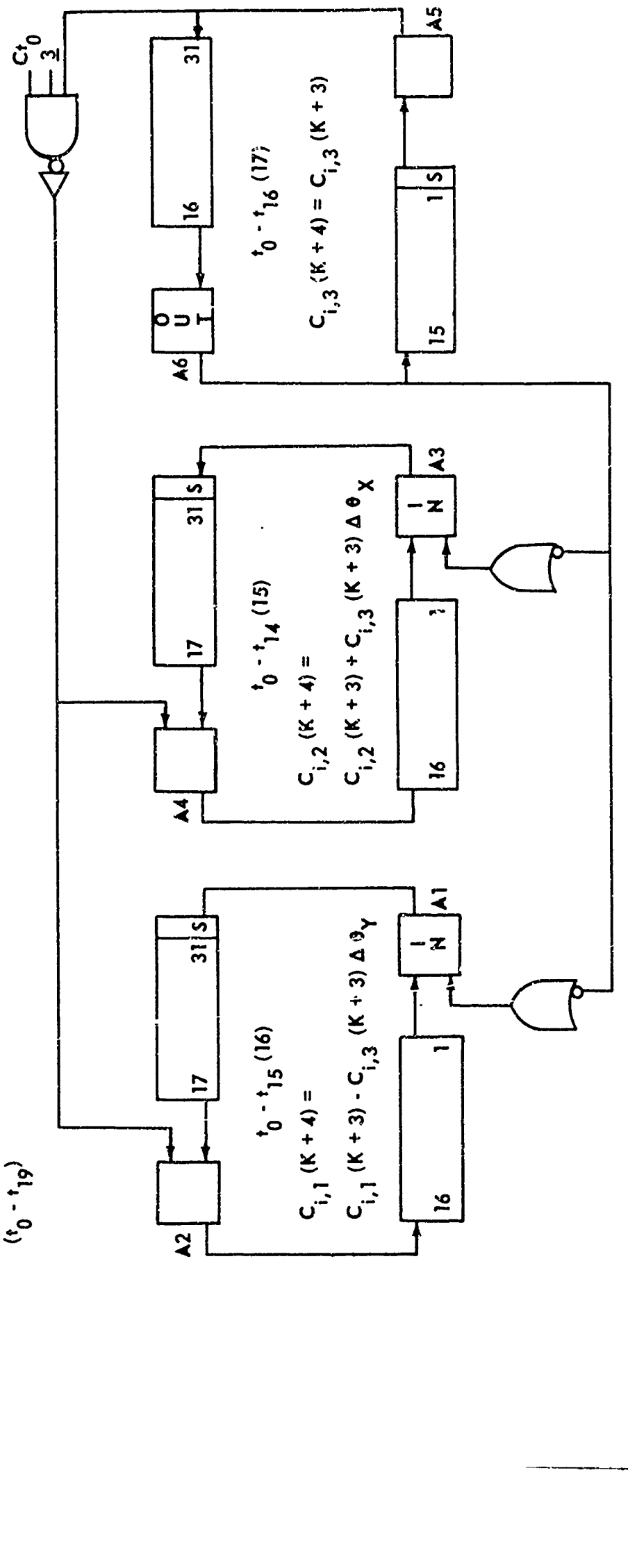

TIMING PERIOD 3
$t_{60} - t_{79} = \underline{3}$
$(t_0 - t_{19})$
A2
17
31
S
$t_0 - t_{15}$ (16)
$C_{i,1}$ (K + 4) =
$C_{i,1}$ (K + 3) - $C_{i,3}$ (K + 3) $\Delta\theta_Y$
1
16
IN
A1
A4
17
31
S
$t_0 - t_{14}$ (15)
$C_{i,2}$ (K + 4) =
$C_{i,2}$ (K + 3) + $C_{i,3}$ (K + 3) $\Delta\theta_X$
1
16
IN
A3
A6
16
31
OUT
$t_0 - t_{16}$ (17)
$C_{i,3}$ (K + 4) = $C_{i,3}$ (K + 3)
15
1
S
A5
Ct_0
3

FIGURE 7. TI

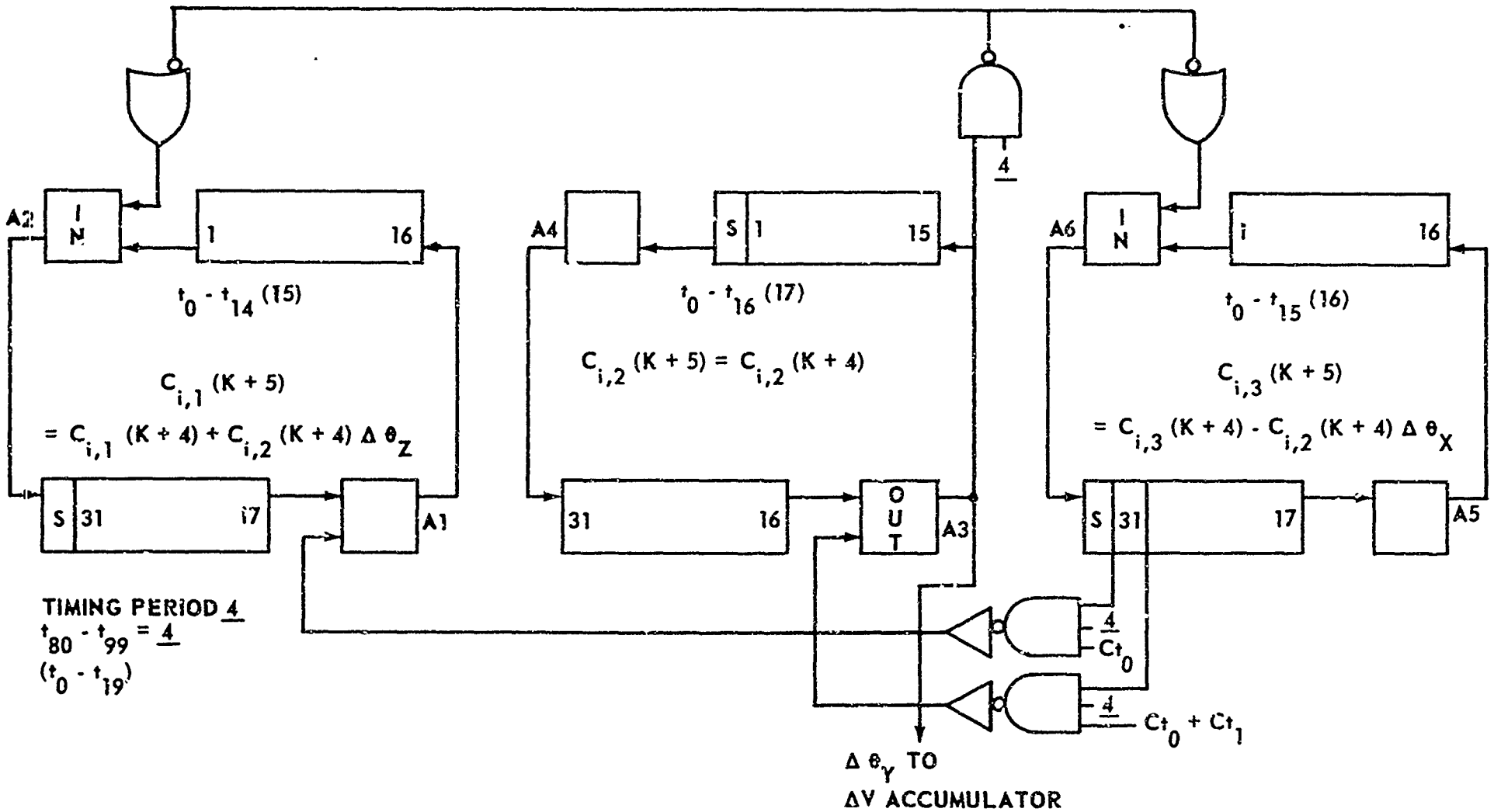


FIGURE 8. TIMING PERIOD 4

TIMING PERIOD 5
$t_{100} - t_{119} = 5$
$(t_{-} - t_{19})$

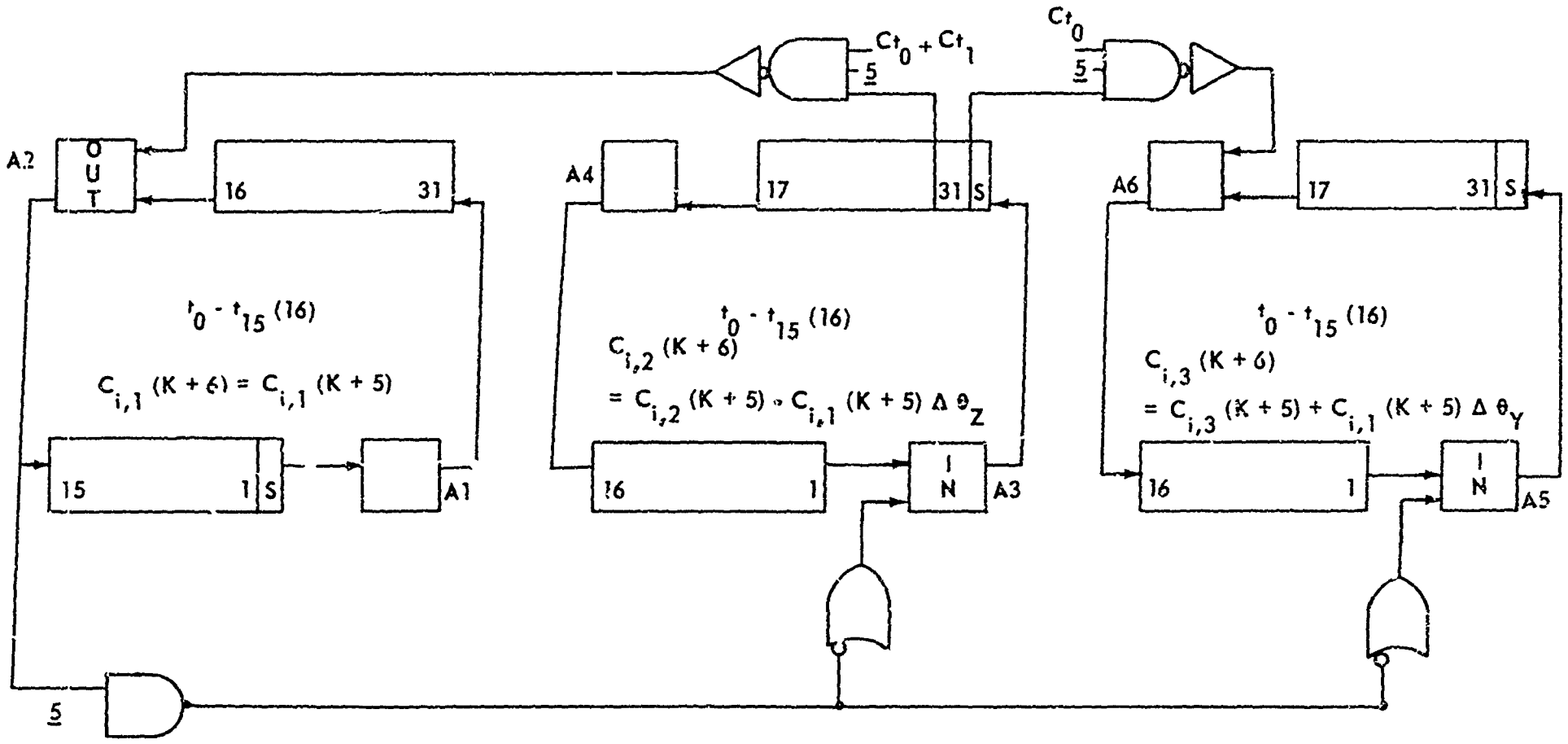


FIGURE 9. TIMING PERIOD 5

into A1 since it will also output during the next timing period. Matrix element 4 receives 16 clock pulses to place bit 1 as the next bit into A4 since it will also receive data during the next timing period. Bits 16 to 31 from ME1 are added to bits 1 to 16 in ME4 through A3. Matrix element 7 receives 16 clock pulses to place bit 1 as the next bit into A6 since it will also receive data during the next timing period. Bits 16 to 31 from ME1 are added to bits 1 to 16 in ME7 through A5.

The overlap from this timing period was described above with timing period 0. That will conclude two successive updatings. Operations will continue by recycling through the six timing periods.

4. Interface With Navigation Computer

The direction cosines and increments of velocity generated in this Digital Differential Analyzer are held in registers during operation. Data are sampled by the navigation computer for solving navigation equations and storage in memory. Logic in the Digital Differential Analyzer is arranged for automatic presetting and readout of registers by the navigation computer.

The D84 PERSHING ground computer will be used for laboratory tests. However, any comparable general purpose computer can be used. The Digital Differential Analyzer contains a buffer register for parallel input or output. For presetting or register readout during testing, the navigation computer or test computer need only be programmed to select a matrix element line for presetting or register readout. The Digital Differential Analyzer will automatically accept incremental gyro and accelerometer outputs and output direction cosines and increments of velocity as required by the navigation computer.

5. Hardware Design

a. Registers

There are eighteen serial registers, each 16 bits long, and six serial 17-bit registers. All are identical with output parallel of the first two most significant bits as required. The eighteen are used in pairs in each of the nine matrix elements and the six are used in pairs in each of the ΔV accumulators. All registers will be built alike and the sequences and external lines used will adapt them to the matrix location or ΔV accumulator where used. The I/O has one 17-bit register with capability for parallel and serial input and output.

b. Adders

There are eighteen identical two-bit adders used in pairs with each pair of the above registers in each of the nine matrix elements. The sequencer and input leads adapt the adder design for use in any matrix location. With the 2's complement system and minor modifications, this adder adds, subtracts, and multiplies input data by 2. The ΔV accumulator also uses a two-bit adder, but its inputs are different enough to justify a slightly different design.

c. Synchronization

The basic computer timing is provided by a 1.2-mHz clock, a recycling counter that counts clock pulses 0 to 19 (20 clock pulses), and a recycling counter that counts 20^{th} clock pulses and goes from 0 to 5 (six timing periods). All timing functions are derived from combinations of these basic counters driven by the clock.

d. Sequencer

The sequencer is shown as a block in Figure 2; however, it is made up of several circuits driven by the counter or clock and working in various combinations to sequence each matrix element through updating and overlap cycles. The sequencing of the three matrix elements in any column is identical for any given timing period; therefore, one set of circuits makes up the sequencer as follows:

(1) Sign Set and Reset and Carry Reset. Updating of a direction cosine occurs only if $\Delta\theta$ occurred and each $\Delta\theta$ has a sign to indicate which way the direction cosine should be updated. The algorithms alternate sign of $\Delta\theta$ to equalize direction cosine updating when there are no body rates; therefore, the sign of $\Delta\theta$ is inverted so that the updating of the direction cosine is accomplished with the proper sign (+) (-) = -, (-) (-) = +.

As pointed out before, the complete direction cosine updating (six word times) consists of two three-word-time updatings. Therefore, each adder inputs twice during the complete updating and always at the same timing period of the six timing period cycle. During the input cycle, logic circuitry sets or resets a flip-flop in the adder when the sign bit is processed through the adder. The flip-flop holds the sign bit through the overlap until the sign bit of the direction cosine is processed. The same logic that resets the sign at that point also resets the carry flip-flop in the adder.

(2) Logic for Initialization and Sign for $\Delta\theta$ and $2\Delta\theta$. The sign of $\Delta\theta$ is held in a series of six flip-flops that are updated once every six timing periods. There is a seventh flip-flop in this logic with a load or lift-off state. The load position inhibits all the adders while the Digital Differential Analyzer is being preset. The lift-off position sets one flip-flop for each of the three axes and resets one flip-flop for each of the three axes. With no gyro motion, the binary pulse rebalance gyros will alternate $\pm\Delta\theta$ and produce a zero net updating of direction cosines. Gyro motions will cause predominance in plus or minus, causing updating in that direction. The logic for $\Delta\theta$ is six flip-flops initialized in the $\Delta\theta$ position. The gyro inputs are sampled every six timing periods and will change to $2\Delta\theta$ when body rates exceed specified limits.

6. Input/Output

The I/O consists of nine flip-flops to select a matrix element for presetting, nine flip-flops to select a matrix element for readout, and three flip-flops to select the ΔV register for readout of the accumulated velocity. A buffer register holds the data that are being preset or read out of the Digital Differential Analyzer. The buffer register is enabled only when one of the 21 flip-flops has been selected. Data are clocked in or out of the buffer once with exactly 17 clock pulses until another flip-flop is selected.

7. Hardware Construction

The Digital Differential Analyzer was designed to be built with large-scale integrated (LSI) circuits. The fact that all the matrix elements are identical, except for slight variations in input signals, offers several levels of logic density concentration that can be implemented with LSI circuits. The design has been partitioned for a breadboard of one row plus controls, to be built out of T^2L integrated circuits components. This will allow laboratory testing and debugging before the whole design is committed to any particular LSI circuit approach. While the breadboard is built and tested, progress of the various LSI circuit approaches is being closely monitored to effect a better selection. The LSI circuit approach to be selected will not necessarily be the one with the highest gate concentration. It is more probable that it will be one with a reasonably high gate concentration, inherent high reliability, low cost, and realistic approach to logistics.

8. Nuclear Effect Considerations

The shift registers making a high portion of the computer are, unfortunately, the most sensitive components to nuclear radiation. The gamma and X-ray radiation will very likely cause change of state. Techniques for hardening are available but not simple. Since the navigation computer will be using all the current data from the Digital Differential Analyzer, it will be more feasible to use the memory in the navigation computer for storage and take advantage of the hardening that must be provided there. After the critical period of a nuclear blast, the information can be reset correctly in the registers. This is the most logical approach since the navigation computer is to provide whatever permanent storage the Digital Differential Analyzer requires and since it also reduces the problem of nuclear hardening to a single solution.

9. Conclusion

The iteration rates to solve direction cosines for a PERSHING II class missile in a strapdown application and the solution of the navigation problem would be prohibitive for a state of the art general purpose machine. However, the high speed, low power, and compact features of LSI circuits offer an excellent approach to build a small serial Digital Differential Analyzer to solve the direction cosine matrix to serve as inputs to the navigation computer. The design of such a Digital Differential Analyzer described in this report lends itself ideally to various levels of LSI circuit construction because of its few different logic circuits used repeatedly as common building blocks, and its serial operation that requires minimum of input and output leads.

This design is unique in that the 3×3 direction cosine matrix is continually updated by merely circulating each word around a pair of adders and registers comprising each matrix element and simultaneously accepting or outputing data to matrix elements in the same string coincident with sampling of gyro inputs. The use of a two-bit adder to add, subtract, and multiply throughout the Digital Differential Analyzer is another advantage of the design. The capability to multiply, by merely adding a flip-flop at the input to the adder, to accommodate random changes in scale factor of gyro inputs, is an extremely simple mechanization when compared to the computer complexity that would be required by the alternate solutions considered. The alternate solution would have made the mechanization of this scheme extremely difficult and maybe prohibitive with present day hardware.

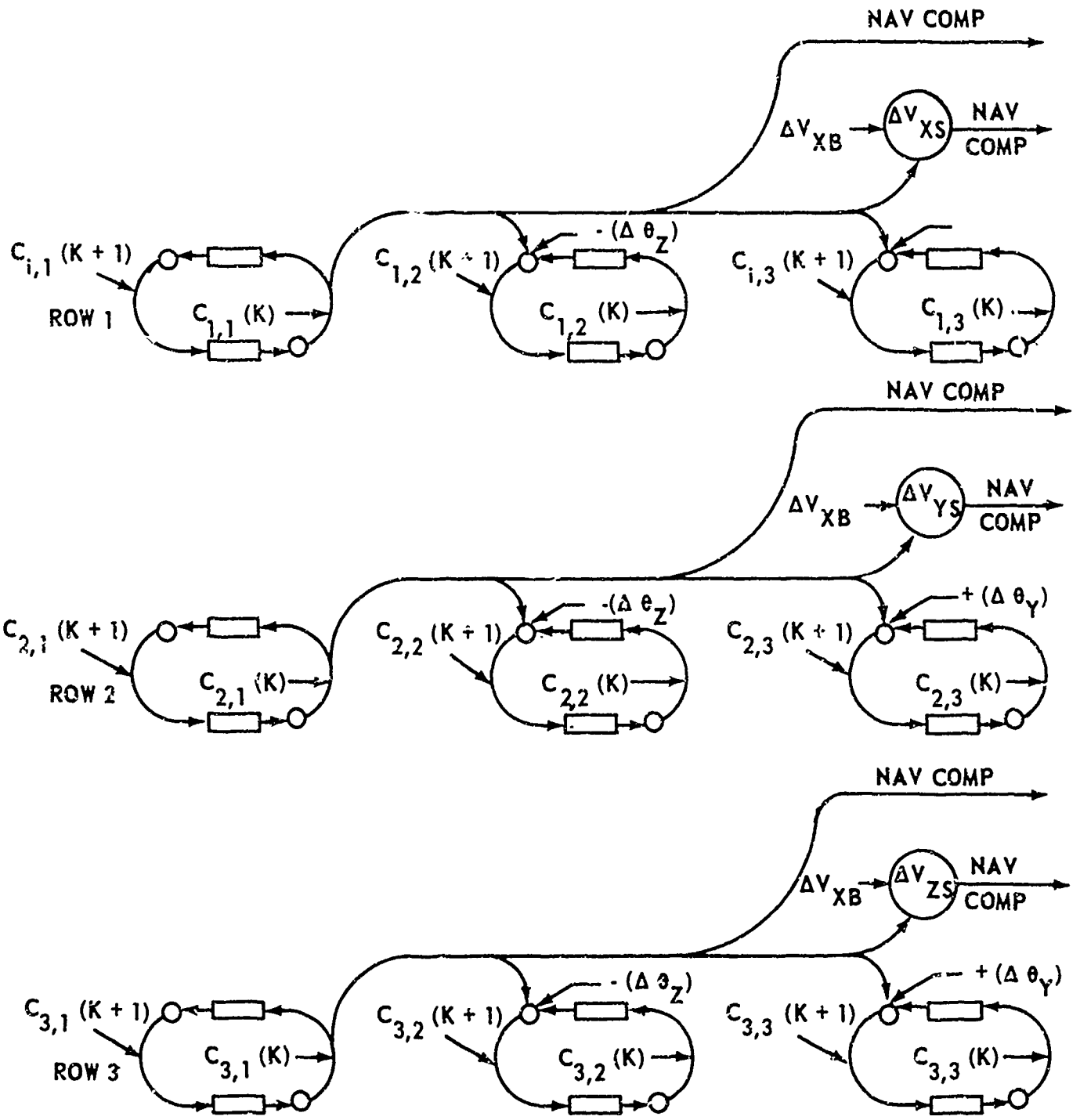
The mechanization of the scheme, as unique as it may appear, did not get by without complications. It would have been very desirable to complete every incremental updating within a single timing period, without overlap. The

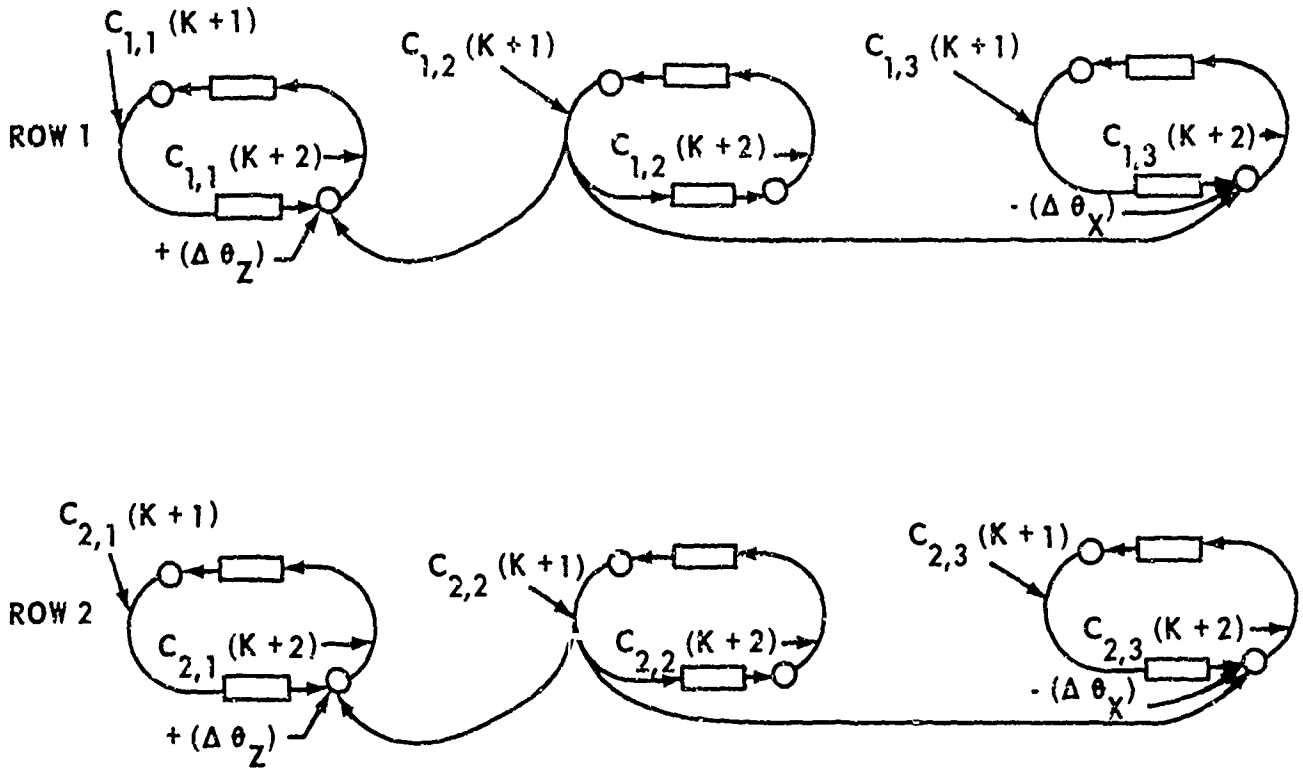
need for overlapping to complete the partial incrementation became apparent early in the design by simulating the flow of data on paper. Overlapping to complete partial updatings created complications because it is not always the same. It requires the computer to store the sign and $\Delta\theta$ or $2\Delta\theta$ of the current and previous complete updating periods (three timing periods); it also requires proper sequencing for each of the columns. Fortunately, the logic to accomplish all the sequencing is fairly simple and is a one-time application in the computer.

Nuclear hardening is recognized as a problem with this computer. The approach selected is to keep the current data from this computer stored in the navigation computer and take advantage of the hardening provided for the navigation computer rather than solving the same problem for the two computers.

The complexities of the computer are more in following the sequence of operations than hardware complexity. The design can be implemented using any of the LSI circuit approaches being developed. It is also feasible to build each row with a different LSI circuit approach to more conclusively evaluate the various more promising LSI circuit approaches.

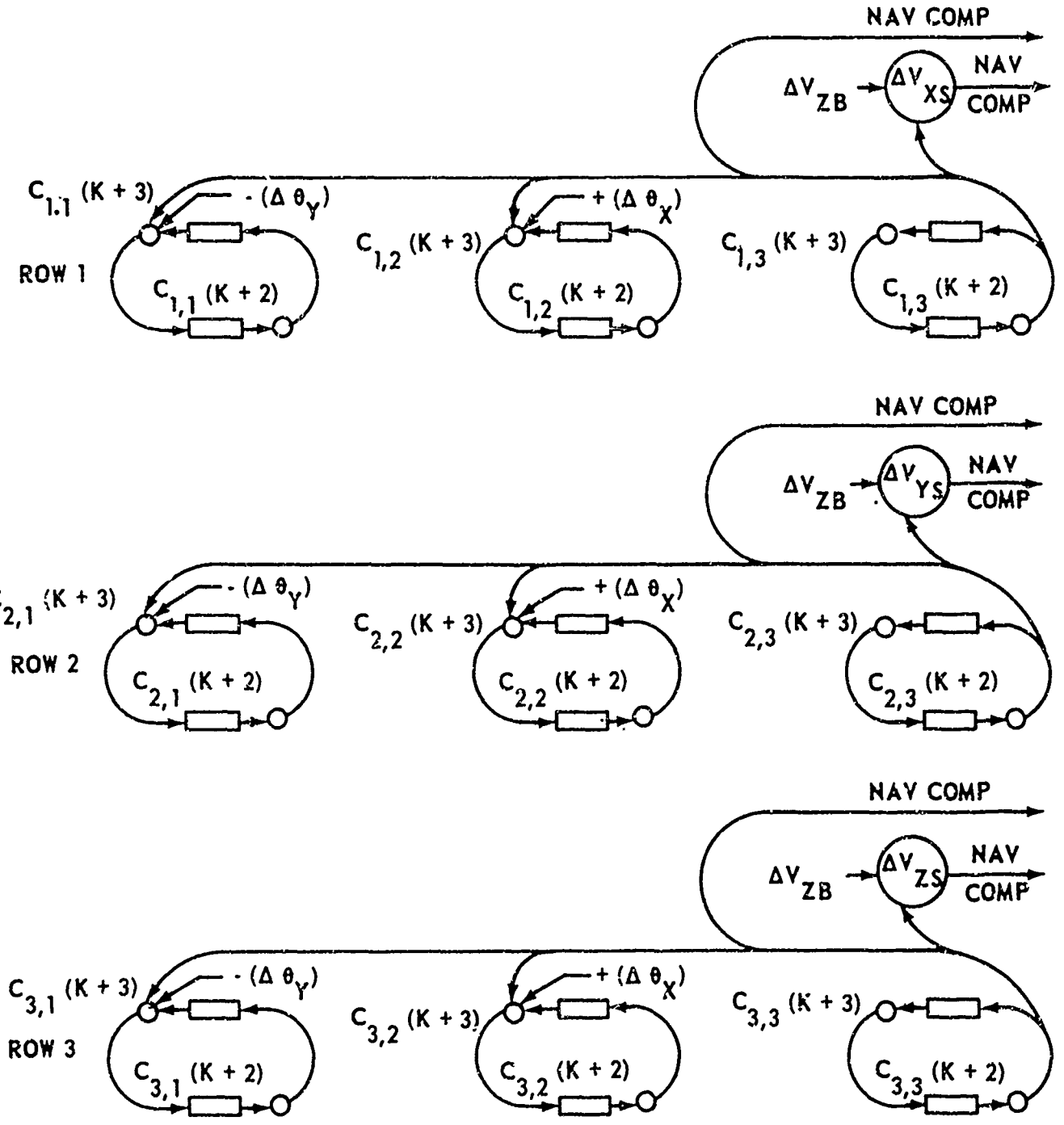
Appendix
FLOW OF DATA IN COMPLETE COMPUTER FUNCTION FOR TWO UPDATINGS

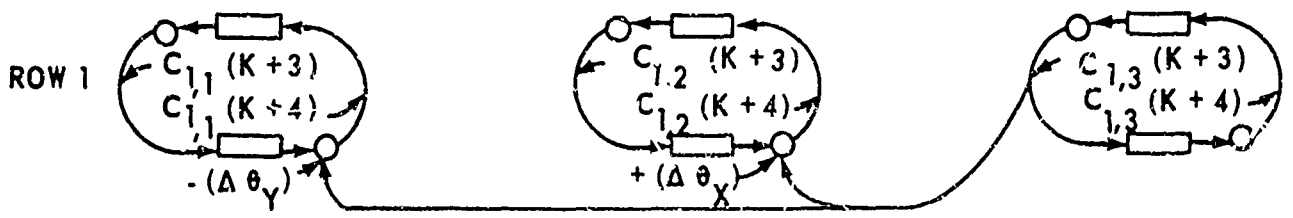

NAV COMP
ΔV XB
ΔV XS
NAV COMP
C i,1 (K + 1)
ROW 1
C 1,1 (K)
- (Δ θ Z)
C 1,2 (K + 1)
C 1,2 (K)
C i,3 (K + 1)
C 1,3 (K)
NAV COMP
ΔV XB
ΔV YS
NAV COMP
C 2,1 (K + 1)
ROW 2
C 2,1 (K)
-(Δ θ Z)
C 2,2 (K + 1)
C 2,2 (K)
+ (Δ θ Y)
C 2,3 (K + 1)
C 2,3 (K)
NAV COMP
ΔV XB
ΔV ZS
NAV COMP
C 3,1 (K + 1)
ROW 3
C 3,1 (K)
- (Δ θ Z)
C 3,2 (K + 1)
C 3,2 (K)
+ (Δ θ Y)
C 3,3 (K + 1)
C 3,3 (K)
CLOCK PULSES
17
15
16
TIMING PERIOD 0

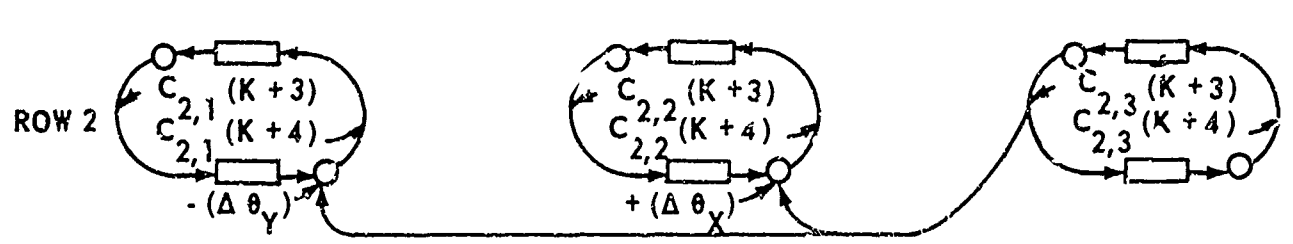


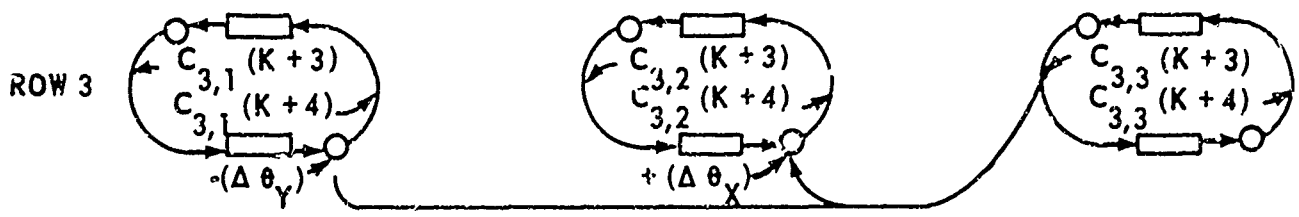
CLOCK PULSES 16 17 15

TIMING PERIOD 1


NAV COMP
ΔV ZB
ΔV XS
NAV COMP
C 1.1 (K + 3)
- (Δ θ Y)
+ (Δ θ X)
C 1,2 (K + 3)
C 1,3 (K + 3)
ROW 1
C 1,1 (K + 2)
C 1,2 (K + 2)
C 1,3 (K + 2)
NAV COMP
ΔV ZB
ΔV YS
NAV COMP
C 2,1 (K + 3)
- (Δ θ Y)
+ (Δ θ X)
C 2,2 (K + 3)
C 2,3 (K + 3)
ROW 2
C 2,1 (K + 2)
C 2,2 (K + 2)
C 2,3 (K + 2)
NAV COMP
ΔV ZB
ΔV ZS
NAV COMP
C 3,1 (K + 3)
- (Δ θ Y)
+ (Δ θ X)
C 3,2 (K + 3)
C 3,3 (K + 3)
ROW 3
C 3,1 (K + 2)
C 3,2 (K + 2)
C 3,3 (K + 2)
CLOCK PULSES
16
16
16
TIMING PERIOD 2

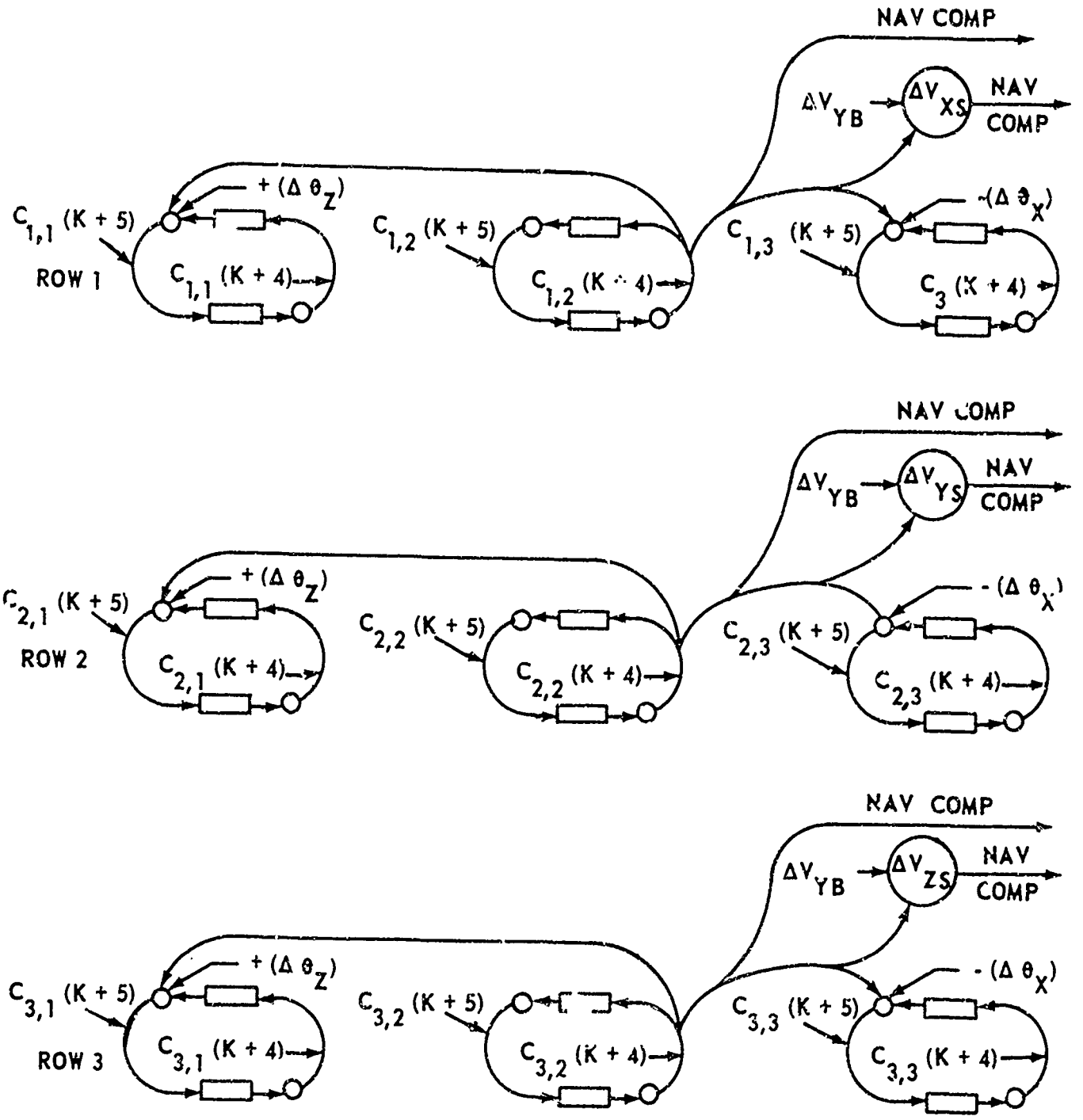


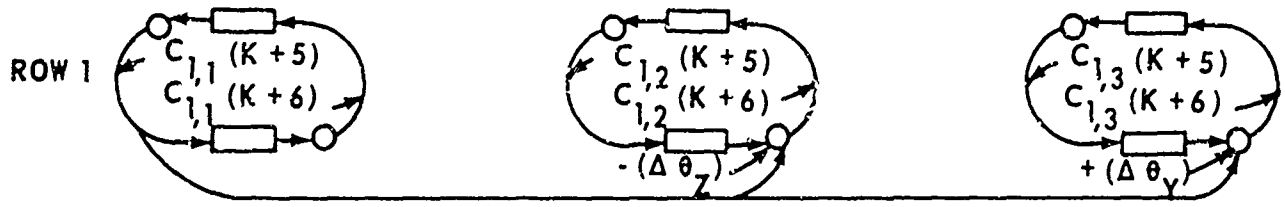


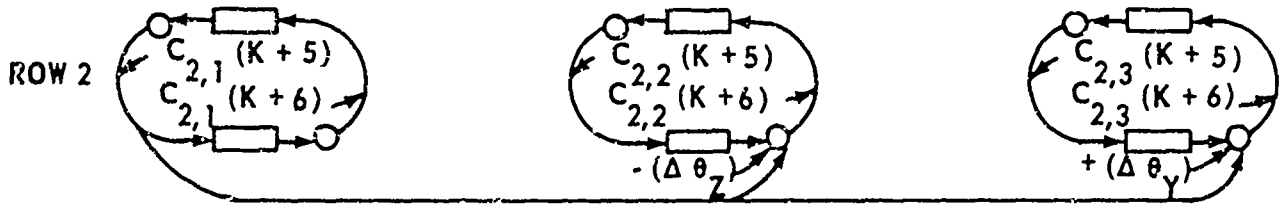


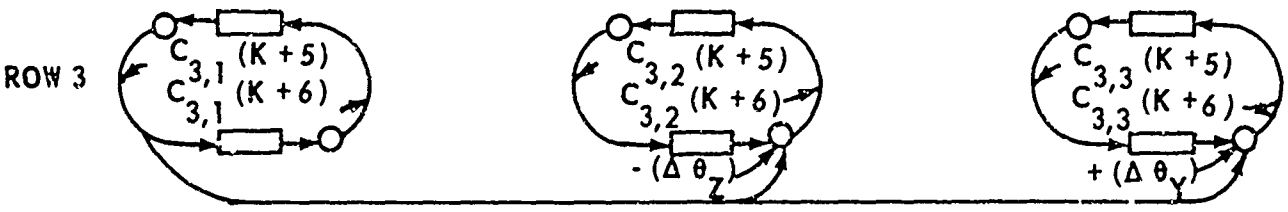
CLOCK PULSES 16 15 17

TIMING PERIOD 3


NAV COMP
ΔV_{YB}
ΔV_{XS}
NAV
COMP
$+(\Delta \theta_Z)$
$-(\Delta \theta_X)$
$C_{1,1}$ (K + 5)
ROW 1
$C_{1,1}$ (K + 4)
$C_{1,2}$ (K + 5)
$C_{1,2}$ (K + 4)
$C_{1,3}$ (K + 5)
C_3 (K + 4)
NAV COMP
ΔV_{YB}
ΔV_{YS}
NAV
COMP
$+(\Delta \theta_Z)$
$-(\Delta \theta_X)$
$C_{2,1}$ (K + 5)
ROW 2
$C_{2,1}$ (K + 4)
$C_{2,2}$ (K + 5)
$C_{2,2}$ (K + 4)
$C_{2,3}$ (K + 5)
$C_{2,3}$ (K + 4)
NAV COMP
ΔV_{YB}
ΔV_{ZS}
NAV
COMP
$+(\Delta \theta_Z)$
$-(\Delta \theta_X)$
$C_{3,1}$ (K + 5)
ROW 3
$C_{3,1}$ (K + 4)
$C_{3,2}$ (K + 5)
$C_{3,2}$ (K + 4)
$C_{3,3}$ (K + 5)
$C_{3,3}$ (K + 4)
CLOCK PULSES
15
17
16
TIMING PERIOD 4







CLOCK PULSES 16 16 16

TIMING PERIOD 5

DISTRIBUTION

	No. of Copies
U.S. Army Missile Command Distribution List A	79
Defense Documentation Center Cameron Station Alexandria, Virginia 22314	20
AMCPM-PE	1
AMSMI-R, Mr. McDaniel	1
-RG	1
-RGX	1
-RGC	1
-RGN	1
-RGE	1
-RGI	1
-RGL	1
-RGG	10
-RBL	5
-RPR	1

UNCLASSIFIED

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author)	2a. REPORT SECURITY CLASSIFICATION
Army Inertial Guidance and Control Laboratory and Center Research and Development Directorate U. S. Army Missile Command Redstone Arsenal, Alabama 35809	Unclassified
	2b. GROUP N/A

3. REPORT TITLE

LARGE-SCALE INTEGRATED CIRCUIT COMPUTERS FOR COORDINATE TRANSFORMATION

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

5. AUTHOR(S) (First name, middle initial, last name)

Raul Pena, Jr.

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
12 August 1968	37	0

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
b. PROJECT NO. (DA) 1X27919D678 AMC Management Structure Code No.	RG-TR-68-12
c. 5282.12.12700	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d.	AD

10. DISTRIBUTION STATEMENT

This document is subject to special export controls and each transmittal to foreign governments or foreign nationals may be made only with prior approval of this Command, ATTN: AMSMI-RG

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
None	Same as No. 1

13. ABSTRACT

Methods and algorithms for navigation systems with body-mounted inertial instruments have been studied by various research groups throughout the country. The high iteration rates for coordinate transformation in a PERSHING II missile and the complexity of the navigation solutions make a general purpose computer or special purpose computer unfeasible for this application. A scheme to use a small Digital Differential Analyzer as the front end of a general purpose navigation computer to perform the coordinate transformation is feasible.

This report describes a unique mechanization of a Digital Differential Analyzer to transform the outputs from inertial instruments from a body-fixed coordinate system to a stabilized coordinate system. The design is specially adaptable to various levels of logic density using large-scale integrated circuits because it has few different logic circuits used repeatedly and performs serial computation that requires minimum input and output leads. Unique features of the Digital Differential Analyzer are that it solves a 3×3 direction cosine matrix by merely circulating data around a pair of registers and adders for each matrix element and uses a two-bit adder to add, subtract, and multiply by $(2)^{n}$.

DD FORM 1 NOV 68 1473 REPLACES DD FORM 1473, 1 JAN 64, WHICH IS OBSOLETE FOR ARMY USE.

UNCLASSIFIED

Security Classification

14. KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Body-mounted inertial systems						
-navigation systems						
-algorithms						
Digital Differential Analyzer						
Navigation computer						
Large-scale integrated circuits						